Available online at www.sciencedirect.com

ScienceDirect

Procedia Social and Behavioral Sciences 20 (2011) 826–835

Procedia
Social and Behavioral Sciences

ELSEVIER

14th EWGT & 26th MEC & 1st RH

A stochastic programming approach for robust vehicle scheduling in public bus transport

Marc Naumann[a,*], Leena Suhl[a], Stefan Kramkowski[a]

[a] *Decision Support & Operations Research Lab, University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany*

Abstract

We present a new stochastic programming approach for robust vehicle scheduling in public bus transport. Our approach uses typical disruption scenarios during the optimization to minimize the expected sum of planned costs and costs caused by disruptions. The schedule is represented as a time-space network with all connecting arcs to enable independent penalization of every connection between two consecutive service trips. Our method significantly decreases total expected costs compared to just minimizing planned costs and outperforms a simple approach of adding fixed buffer times between service trips. Despite the increased computational complexity, small and medium-sized real-world instances can be solved.

Selection and/or peer-review under responsibility of the Organizing Committee.

Keywords: vehicle scheduling; public bus transport; stochastic programming; robust optimization; disruptions

1. Vehicle scheduling in public transport with disruptions

The vehicle scheduling problem is one of the operational planning problems in public transport. Usually, the lines and their frequencies are planned by the local authority. Then a tendering takes place, and public transport companies can obtain a license for lines or line-bundles.

Next, the public transport company plans the schedule for its vehicles considering vehicle types as well as vehicles which transport the passengers. The plans must be coherent as well as resource-efficient which makes this planning step a not easy task.

The following step is the crew scheduling. The vehicle schedule is first divided into anonymous tasks which are then grouped into anonymous duties for a day.

* Corresponding author. Tel.: +49-5251-60-5263; fax: +49-5251-60-3542
E-mail address: naumann@dsor.de

Selection and/or peer-review under responsibility of the Organizing Committee
doi:10.1016/j.sbspro.2011.08.091

The crew rostering finally concatenates the anonymous day-duties to weekly duties, which are then assigned to individual drivers. An integrated consideration of these planning phases leads to a very high complexity so that sequential planning is practical. Figure 1 gives an overview over the operational planning process in public transport. (Compare Huisman et al. (2004))

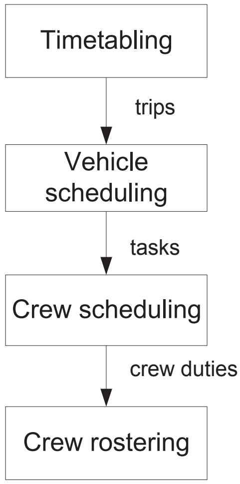

Figure 1 Operational planning process in public transport

The vehicle schedules are traditionally planned several weeks before their execution. The buses are assigned to the given timetabled trips, so that every trip is covered by a bus. Thereby the trip has to be carried out by a vehicle type allowed for this trip, and the vehicles have to start and end at the same depot. The objective is usually to minimize planned costs, which consist of fixed cost per vehicle, variable cost per driven distance and time spent outside the depot.

On the day of operations, disruptions can occur and cause delays which imply increased operational cost and penalty fees. Huisman et al. (2004) use a quadratic function to penalize larger delays overproportionally. If there is waiting time between two service trips, delays can be absorbed. But as buffer time in schedules cause more planned cost, cost-optimal schedules tend to contain few buffers that can absorb disruptions.

Before describing how we aim to increase robustness of vehicle schedules, we define some terms.

A *primary delay* is a delay that is directly caused by a disruption, for example if a road is blocked by congestion, etc. As disruptions occur, primary delays cannot be avoided. Primary delays cause a late arrival of a service trip that has departed on time.

If a delayed service trip causes a delayed start of a following service trip, we call this delay a *secondary delay*. Secondary delays occur because of dependencies of consecutive service trips. They can be prevented by introducing buffer time. Most public transport companies, especially smaller companies, do not have an operations control center so that they cannot dynamically react on disruptions with re-planning. Therefore, if not enough buffer time can absorb the delayed arrival of a service trip, the subsequent service trip will start later. This effect is called *delay propagation*.

We call a vehicle schedule more *delay tolerant*, if it is able to absorb secondary delays better than a reference schedule. A schedule that has less secondary delays is usually more expensive, but causes less penalty costs. Our goal is to optimize the expected sum of planned costs plus additional operational costs caused by disruptions, plus penalty costs, so that we gain a schedule that has the lowest expected costs over all delay scenarios. In this paper, we call a schedule more *robust*, if the delay tolerance of the schedule is higher.

To the best of the authors' knowledge, there is no literature on stochastic optimization models for robust vehicle scheduling in public bus transport. Bunte and Kliewer (2009) give an overview on general vehicle scheduling models in public transport. Huisman et al. (2004) solve the dynamic vehicle scheduling problem where they use scenarios for travel times to consider disruptions and robustness. Dessouky et al. (1999) present a summary of distribution functions used for delays in public transport in former studies.

The vehicle scheduling problem in public transport is related to the aircraft routing problem. As differences, the aircraft routing does not integrate the assignment of the typeclass and no deadheads are allowed, but the planning horizon is usually longer. In this field many optimization models have been developed, so that some like Lan et al. (2006) could be adjusted to the vehicle scheduling problem in public bus transport under uncertainty.

2. Delays and penalty costs in network models

Dessouky et al. (1999) define lateness as a deviation from scheduled arrival time, which corresponds to our definition of a delay. The distribution function used for lateness in former studies was the exponential distribution. We use the exponential distribution and extend it with a factor depending on the daytime of the service trip to consider the impacts of rush hours.

We consider primary delays on service trips, but delays on depot-trips and deadhead-trips are not considered (like in Huisman et al. (2004)). The scenarios are generated in a way that they range from scenarios with a very low probability for a delay and a low delay length to scenarios with a high probability and a high delay length. The reason for generating scenarios in this way is that we want to cover very bad days, such as days in winter with bad weather conditions or days with road closures downtown, as well as days where only few disruptions occur.

To obtain one scenario, we draw a random value for each service trip from the delay distribution. This random value is either zero if the service trip is not delayed or is the delay length for the service trip. Therefore we approximate the original distribution with n values for every service trip by generating n scenarios.

Although we have generated the scenarios in a realistic way, the best would be using real delay data of past days. This would regard the characteristics of the particular road network and also be the most convenient way in practice.

A service trip is primarily delayed, if it has started on time and arrives not punctually at its ending station. In our model we consider penalty cost, if a primarily delayed service trip causes a non-punctual start of a following service trip. Therefore, we add penalty costs to the connection of the delayed service trip with its following service trips, if the buffer time is too small to compensate the delay.

For modeling we use a time-space network (TSN) like in Kliewer et al. (2006), but we have to regard more deadhead and waiting arcs, because of the previous mentioned delay penalty costs on connections. Let us consider a part of a time-space network at a certain bus station, as shown in Figure 2.

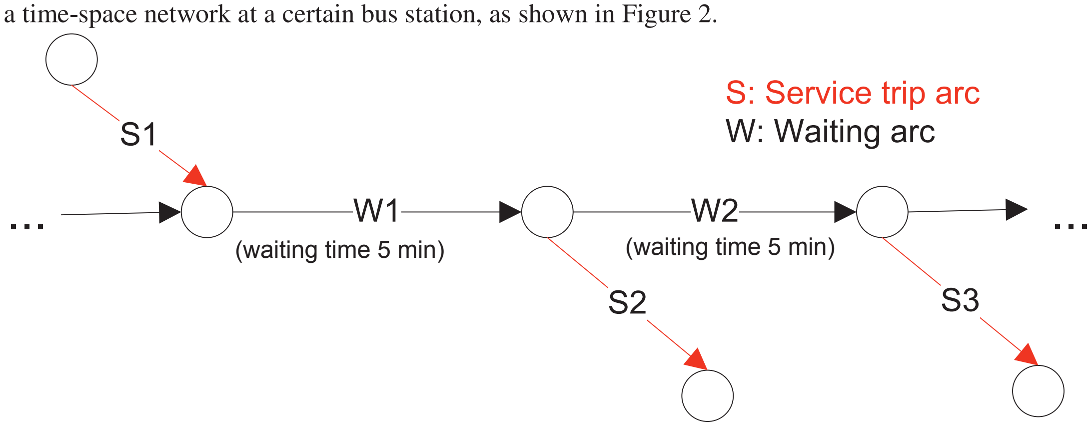

Figure 2 Penalty cost in a time-space network

The service trip S1 ends at this bus station, and S2 and S3 start 5 minutes and 10 minutes later at the same bus station. Now we assume that service trip S1 has a delay and arrives at the bus station 8 minutes later. As a consequence, S2 cannot depart on time when the same bus is used for it, because there is only a planned waiting time of 5 minutes. Therefore penalty costs are added to the waiting arc W1.

A more robust plan could now decide to use the same bus for S1 and S3 and use another bus for S2 to avoid the delayed start of S2. S3 could start punctually because of the 10 minute-buffer.

A problem of the TSN is now that if S3 follows S1, the arcs W1 and W2 would be used to connect these service trips. Using W1 and W2 would absorb the delay of 8 minutes, but the penalty costs of W1 are included anyway. Therefore we need an additional waiting arc beginning at the end-node of S1 and ending at the start-node of S3.

For the planned cost of the waiting arcs, the equality

$$planned\ cost(W1) + planned\ cost(W2) = planned\ cost(W3) \quad (1)$$

is satisfied, but

$$penalty\ (W1) + penalty(W2) \neq penalty(W3) \quad (2)$$

is not satisfied. Because of this non-additivity of the penalty costs, the transitivity of waiting arcs in a TSN cannot be utilized anymore. Therefore we use a TSN with all connecting arcs for our model to consider penalty costs.

We now describe the integration of delays in such a network. There are five possible types of arcs connecting two service trip arcs:

- A waiting arc connects them, if the service trips end and start at the same bus station at different times. (W1 in Figure 3)
- If they end and start at the same time and station, a waiting arc with waiting time 0 connects them. (W2)
- An additional deadhead arc is used to model a deadhead to the depot, waiting time in the depot and a deadhead back to the same bus station, if there is enough time to do this. (DH1)
- A deadhead arc is used, if the two service trips end and start at different stations and there is enough time to connect them. (DH2)
- An additional deadhead arc is again used to model a deadhead to the depot, waiting time in the depot and a deadhead to the bus station where the next service trip starts, if there is enough time. (DH3)

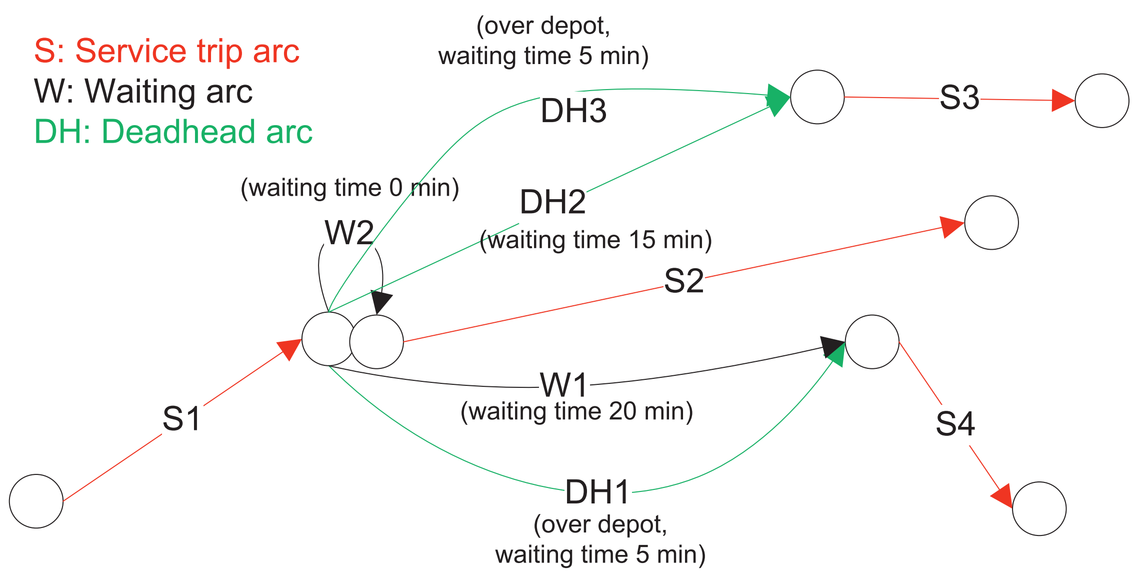

Figure 3 Penalty cost in a time-space network with all connecting arcs

Now, it is possible to penalize every connection independently. For example, if S1 arrives with a delay of 8 minutes, the arcs W2, DH1 and DH3 can be penalized while DH2 and W1 are not penalized. Note that no arc can be excluded in advance because it can be dominated by costs of another one, as their costs depend on the delay scenarios. With the stochastic programming model shown in the next chapter, we aim to minimize the total expected costs consisting of planned costs and disruption costs. We also show the tradeoff between planned costs and disruption costs by restricting one of these cost-components and optimizing the other component.

3. Mathematical optimization model

We use the following deterministic model formulation for the vehicle scheduling problem as a basis for further development. For an overview on vehicle scheduling problems see Bunte and Kliewer (2009).

Sets:

NL	Set of network layers
F	Set of service trips
ES_f	Set of all service trip arcs representing service trip f
V_{nl}	Set of nodes in network layer nl
E_{nl}	Set of arcs in network layer nl

Parameters:

c_e	Cost of arc e
va_e	Beginning-node of arc e
ve_e	Ending-node of arc e

Variables:

x_e	Flow of arc e

Objective function:

$$\min \sum_{nl \in NL} \sum_{e \in E_{nl}} x_e \cdot c_e \qquad (3)$$

Flow-conservation constraints:

$$\sum_{i \in E_{nl} | va_i = v} x_i - \sum_{i \in E_{nl} | ve_j = v} x_j = 0 \qquad \forall v \in V_{nl}, nl \in NL \qquad (4)$$

Cover constraints:

$$\sum_{e \in ES_f} x_e = 1 \qquad \forall f \in F \qquad (5)$$

Integrality constraints:

$$x_e \in Z \qquad \forall e \in E_{nl}, nl \in NL \qquad (6)$$

Bounds:

$$l_e \leq x_e \leq u_e \qquad \forall e \in E_{nl}, nl \in NL \qquad (7)$$

In the following this model is extended to a stochastic programming vehicle scheduling model. Like Huisman et al. (2004), we use a quadratic penalty function to penalize larger delays overproportionally. The costs of one delay of α seconds should be as high as the fixed costs of one vehicle for one day. Therefore the penalty costs for arc e, which is one of the five arc types between two service trips described above, in scenario s are:

$$penalty_{s,e} = y_{s,e}^2 \cdot \frac{c_{nl}^{fix}}{\alpha^2} \qquad (8)$$

$y_{s,e}$ is the starting delay of the service trip following on connection arc e in scenario s or zero if e is not used. The paramenter c_{nl}^{fix} denotes the fixed costs for the usage of one bus for one day of the bus-type in network-layer nl. The objective function now reads:

$$\min \sum_{nl \in NL} \sum_{e \in E_{nl}} c_e \cdot x_e + \frac{1}{|S|} \sum_{nl \in NL} \sum_{e \in E_{nl}} \sum_{s \in S} \left(y_{s,e}^2 \cdot \frac{c_{nl}^{fix}}{\alpha^2} \right) \quad (9)$$

The set S is the scenarioset. The penalty costs for a connection arc only have to be considered, if the arc has a flow greater than 0. Therefore the following constraints are added.

$$y_{s,e} = d_{s,e} \cdot x_e \qquad \forall s \in S, e \in E_{nl}, nl \in NL \quad (10)$$

The parameter $d_{s,e}$ is the starting delay of the service trip following on connection arc e in scenario s. It is the maximum of 0 and the primary delay of the service trip preceding e minus the buffer time of arc e.

Unfortunately this model is a quadratic optimization model and therefore computationally hard. But as the flow of each arc, except the circulation arc, is 0 or 1 and the circulation arc will not have any penalty costs, we can reformulate the model as an equivalent linear model. We plug the equation $y_{s,e} = d_{s,e} \cdot x_e$ in the objective function and obtain:

$$\begin{aligned}
&\min \sum_{nl \in NL} \sum_{e \in E_{nl}} c_e \cdot x_e + \frac{1}{|S|} \sum_{nl \in NL} \sum_{e \in E_{nl}} \sum_{s \in S} \left(x_e^2 \cdot \frac{d_{s,e}^2 \cdot c_{nl}^{fix}}{\alpha^2} \right) \\
\xRightarrow{x_e \in \{0,1\}} &\min \sum_{nl \in NL} \sum_{e \in E_{nl}} c_e \cdot x_e + \frac{1}{|S|} \sum_{nl \in NL} \sum_{e \in E_{nl}} \sum_{s \in S} \left(x_e \cdot \frac{d_{s,e}^2 \cdot c_{nl}^{fix}}{\alpha^2} \right) \\
= &\min \sum_{nl \in NL} \sum_{e \in E_{nl}} x_e \cdot \left(c_e + \frac{1}{|S|} \sum_{s \in S} \frac{d_{s,e}^2 \cdot c_{nl}^{fix}}{\alpha^2} \right) \quad (11)
\end{aligned}$$

The reformulated stochastic optimization model now reads:

Objective function:

$$\min \sum_{nl \in NL} \sum_{e \in E_{nl}} x_e \cdot \left(c_e + \frac{1}{|S|} \sum_{s \in S} \frac{d_{s,e}^2 \cdot c_{nl}^{fix}}{\alpha^2} \right) \quad (12)$$

Flow-conservation constraints:

$$\sum_{i \in E_{nl} | va_i = v} x_i - \sum_{i \in E_{nl} | ve_j = v} x_j = 0 \qquad \forall v \in V_{nl}, nl \in NL \quad (13)$$

Cover constraints:

$$\sum_{e \in ES_f} x_e = 1 \qquad \forall f \in F \quad (14)$$

Integrality constraints:

$$x_e \in Z \qquad \forall e \in E_{nl}, nl \in NL \quad (15)$$

Bounds:

$$l_e \leq x_e \leq u_e \qquad \forall e \in E_{nl}, nl \in NL \quad (16)$$

4. Results

In this chapter, we show the results for the vehicle schedules calculated with the above stochastic programming approach. We compare it with a cost-optimal vehicle schedule and a simple approach that adds fixed buffer times after each service trip. If a service trip is delayed, these buffer times are used to absorb (at least a part of) the delays. The timetable used for our calculations is a real instance of a German city with 426 service trips.

At first, we show the tradeoff between planned costs and penalty costs. We expect that there are solutions with low planned costs, but high penalty costs as well as with low penalty costs, but high planned costs. The value of α is 1800, so that a delay of 30 minutes is as expensive as using one additional bus. Figure 4 shows the tradeoff between planned costs and penalty costs.

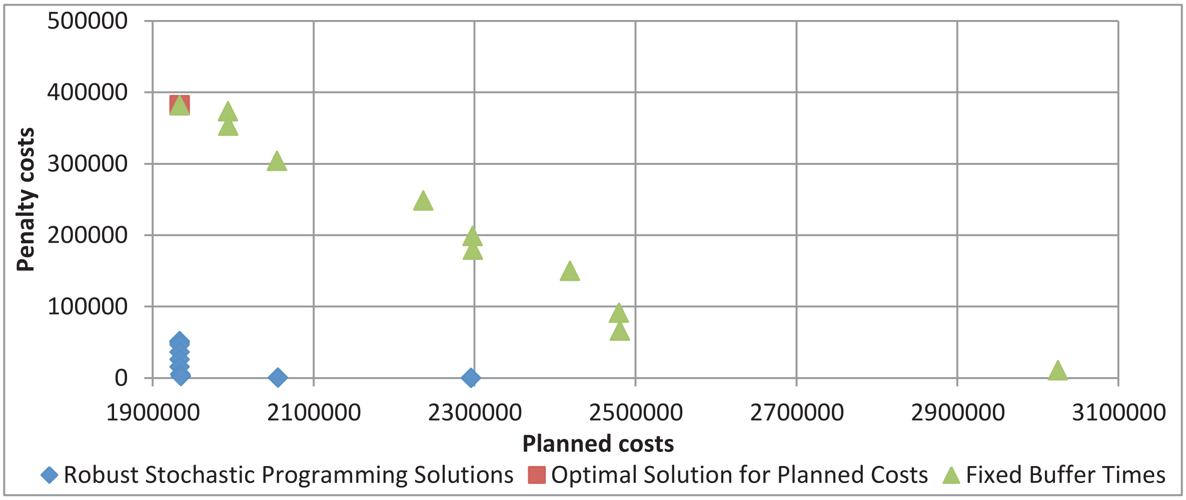

Figure 4 Tradeoff: planned and penalty costs

The figure shows that the optimal solution for planned costs has the highest penalty. It also comes out that the additional buffer times cause a decrease in penalty costs, but they significantly increase the planned costs. This happens for all different amounts of buffer times. These solutions are calculated with buffer durations of 15, 30, 60, 120, 180, 240, 300, 420, 600 and 1200 seconds. The buffers are added after each service trip and then the planned costs are minimized.

The solutions calculated with stochastic programming form a front of pareto-optimal solutions and dominate the simple approach with fixed buffer times. They are calculated by restricting the penalty costs. The penalty costs can be decreased to very small values with a slight increase in planned costs. Table 1 shows a solution comparison.

Table 1 Solution comparison

Solution Approach	Total Costs	Planned Costs	Penalty	#Vehicles
Minimize planned costs	2315517	**1933416**	382100	**32**
Stochastic programming	1979398	**1933416**	45981	**32**
Stochastic programming	1969436	1933424	36012	**32**
Stochastic programming	1959322	1933487	25835	**32**
Stochastic programming	1949175	1933661	15514	**32**
Stochastic programming	1939493	1934322	5171	**32**
Stochastic programming	**1937573**	1934988	2586	**32**
Stochastic programming	2056077	2055563	514	34
Stochastic programming	2296140	2296140	**0**	38
Fix Buffer Time 15s	2367236	1993618	373619	33
Fix Buffer Time 30s	2347397	1994178	353219	33
Fix Buffer Time 60s	2358511	2054656	303856	34
Fix Buffer Time 1200s	3036038	3025127	10911	50

The table proofs that all the stochastic programming solutions are better in total costs than the other solutions. This could have been expected for at least one SP-solution, because stochastic programming always finds the optimal solution under the given data. Moreover, most of the solutions do not use more busses than the optimal solution for planned costs. This is very good in practice, because companies are not always willing to increase the number of used vehicles.

In our preliminary results, the penalty costs are calculated when a disruption causes a delayed start of a following service trip. Further delay propagation was not considered. Under these assumptions, stochastic programming finds the optimal solution. Now, we evaluate the different vehicle schedules with a simulation tool that considers further delay propagation. Our solution approach now becomes heuristic, because it does not consider the impacts of entire delay propagation.

Because of not considering entire delay propagation in the stochastic optimization model, the impacts of disruptions and the resulting costs caused by disruptions are underestimated. To compensate this, we change the parameter α of the optimization model to other values. This means that we change the impact of delays on the penalty costs. If we choose a lower value for α, a smaller delay will cause penalty costs in the amount of the fixed costs for the usage of one bus for one day. We therefore overestimate the penalty costs of a delayed start of a service trip to compensate the underestimation because of the lack of delay propagation in the optimization model.

After having calculated the vehicle schedules with different values for α, we use a simulation software to evaluate the vehicle schedules with the starting value of α, which is 1800 seconds. Figure 5 shows the result.

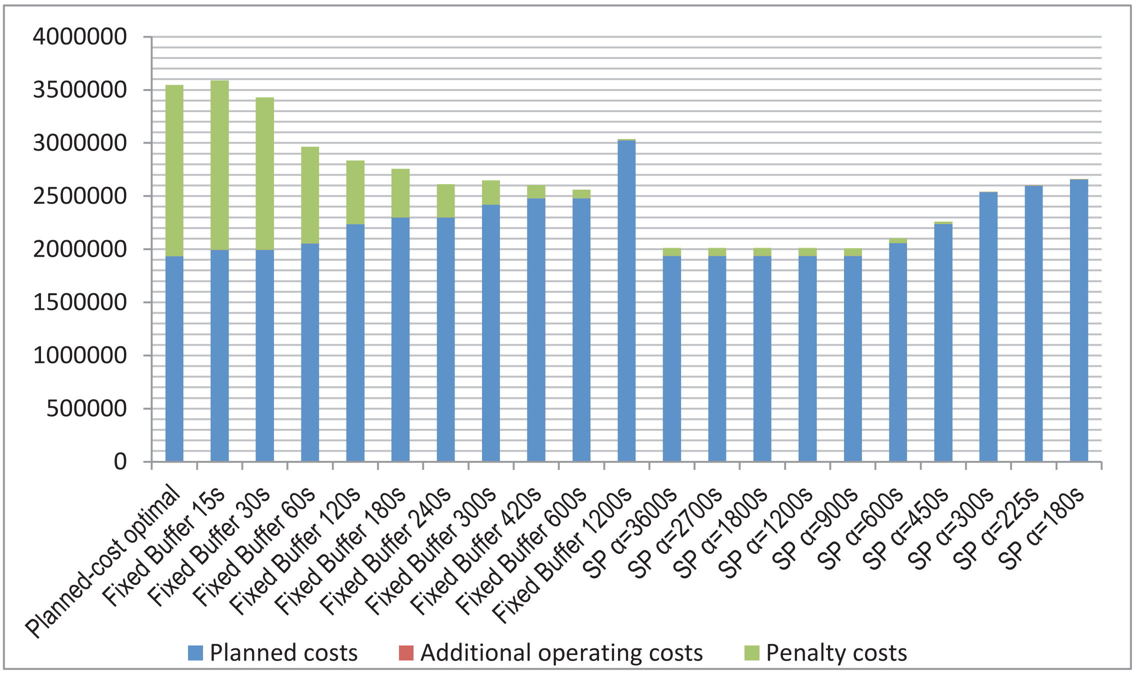

Figure 5 Costs of vehicle schedules with different α

The first observation is that the penalty costs for the planned-cost optimal solution are significantly higher when we consider entire delay propagation. Because of this fact adding fixed buffer times between the service trips now can decrease the total costs. This was not possible in our preliminary results. But it again comes out, that adding fixed buffer times between service trips cannot compete with stochastic programming. How should we now choose the value for α? Our results indicate that the values 2700, 1800, 1200 and 900 produce good solutions for this timetable. (1800 is the value that was used in the simulation tool to calculate the penalty with complete delay propagation.)

Figure 6 shows the average changes in total costs of four different real timetables depending on the method used to create the vehicle schedule.

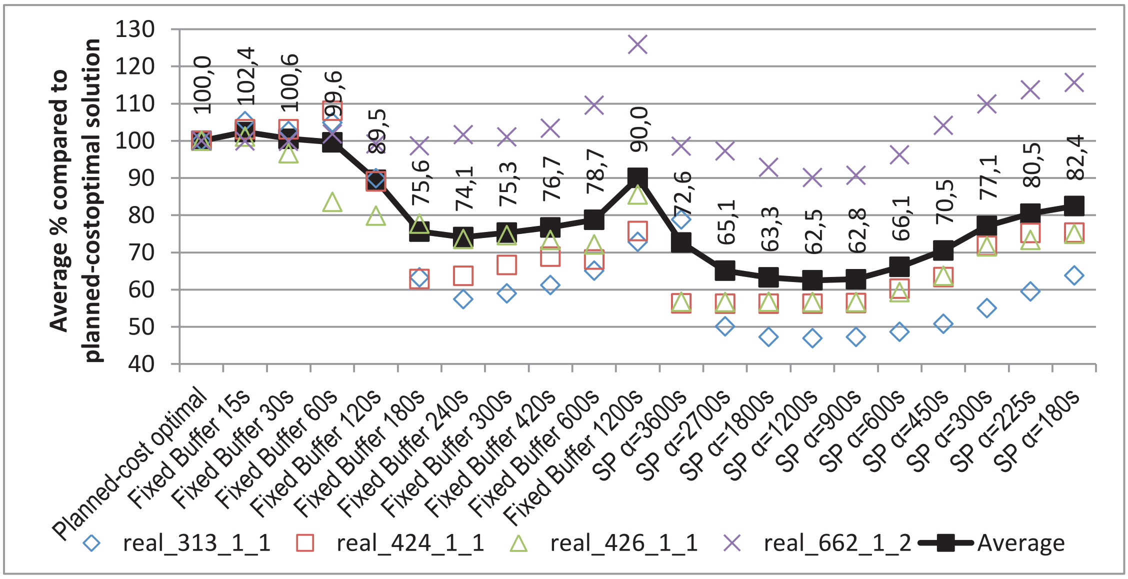

Figure 6 Changes in total costs

The result that an α of 1800 produces good results shows that our stochastic programming solution is applicable even though it does not consider entire delay propagation. The best results can be obtained with an α of 1200. Thus, a small overestimation of the delay costs in the optimization model is the best choice to compensate the heuristic delay propagation.

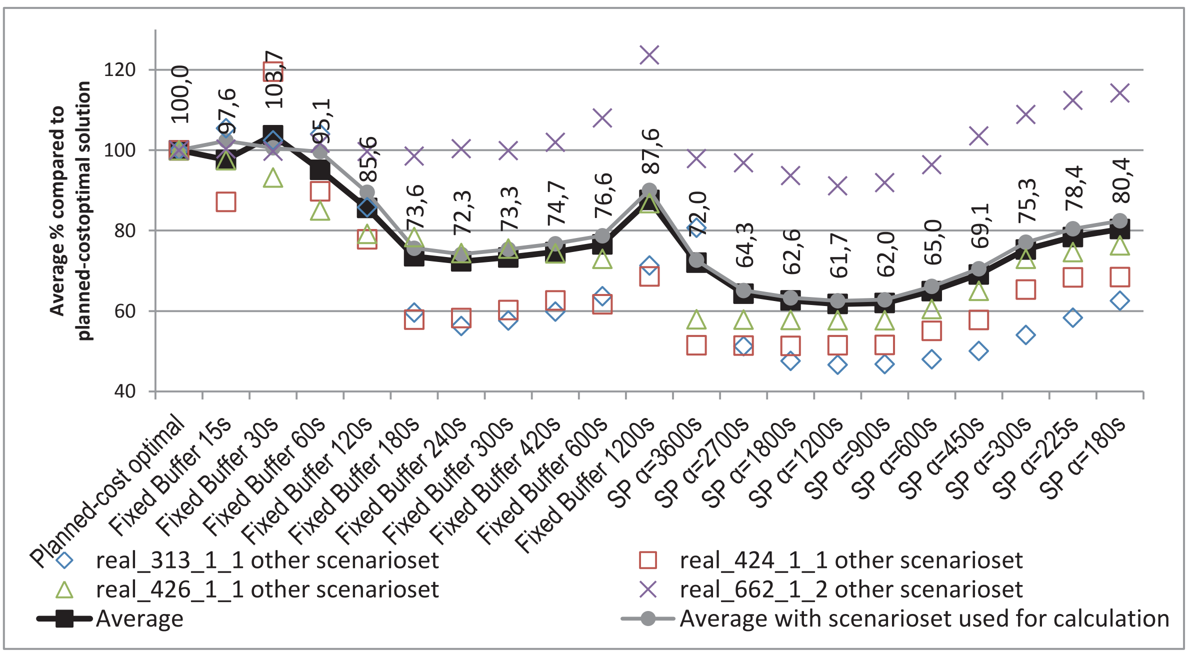

Figure 7 Evaluation with different scenariosets

Because the real instances used to calculate these test results contain a large number of service trips that can be delayed, we cannot calculate every possible combination of delayed service trips and use them as scenarios. Therefore 100 scenarios were included into the optimization model, and the model was optimized with this data. It is now important to show that the vehicle schedule is not only good because the solutions fit to the specific scenarios with which the optimization was carried out. The real delays will be different - disruptions will occur on other service trips than in the scenarios. Therefore, we use another scenarioset for evaluation with 300 scenarios. These 300 scenarios also better approximate the original delay distribution for each service trip. Figure 7 shows the results.

It turns out that the other scenarioset does not lead to different conclusions. The stochastic programming solutions are still the best solutions and 1200 is the best value for α. Again 1800, 1200 and 900 are appropriate values for α. We therefore overestimate the penalty costs during the optimization by using an alpha of 1200 instead of 1800 to compensate the lack of entire delay propagation in the optimization model. The simulation shows that the value of 1200 leads to best results. That means that assuming a delay of 1200 seconds would cause the fixed costs for one bus for one day as penalty during optimization leads to the best schedule for the real world where we assume that 1800 seconds delay cause the fixed costs for one bus for one day as penalty. This value for α is appropriate in both cases if we use the same scenarioset for optimization and simulation and if we use different sets.

5. Conclusion and outlook

We have shown that stochastic programming for the vehicle scheduling problem with disruptions leads to superior solutions in terms of total expected costs compared to a simple approach with fixed buffer times. If complete delay-propagation is not considered, our stochastic programming approach finds the optimal solution for the given data, which outperforms the simple approach. We have created a set of pareto-optimal solutions in terms of maximum robustness and minimum planned costs.

When entire delay propagation is considered, the solutions of stochastic programming are still superior in terms of total expected costs compared to the simple approach adding fixed buffer times, although this simple approach now can decrease total costs significantly compared to the planned cost-optimal solutions. A small overestimation of the delay costs in our stochastic optimization model is appropriate to compensate the lack of delay propagation in the optimization model. Using another scenarioset for evaluation confirms the applicability of our approach.

Furthermore, we have shown that the stochastic components and the consideration of a quadratic penalty function do not add significant complexity to the optimization model. This is done with a reformulation of the optimization model and with the calculation of the penalty costs in the network model. This necessitates a network model with an explicit modeling of all connections between service trips, which is computationally more complex than an aggregate time-space network.

But despite the increased computational complexity, real instances can still be solved in reasonable time. For our calculations we have used instances of small-sized German cities with few network layers and with several hundred service trips. Instances of larger cities or large metropolitan areas cannot be solved in reasonable time with our developed optimization model up to now. In future, our model can be integrated with a column generation approach to solve larger instances. When more computational power, more working memory and more efficient solution algorithms for MIPs will be available, this model will be solvable also for timetables of larger cities.

References

Bunte, S. & Kliewer, N. (2009). An Overview on Vehicle Scheduling Models. *Public Transport*, 1, 299-317.

Dessouky, M., Hall, R., Nowroozi, A. & Mourikas, K. (1999). Bus dispatching at timed transfer transit stations using bus tracking technology. *Transportation Research Part C*, 7, 187-208.

Huisman, D., Freling, R. & Wagelmans, A. (2004). A Robust Solution Approach to the Dynamic Vehicle Scheduling Problem. *Transportation Science*, 38, 447-458.

Kliewer, N., Mellouli, T. & Suhl, L. (2006). A time–space network based exact optimization model for multi-depot bus scheduling. *European Journal of Operational Research*, 175, 1616–1627.

Lan, S., Clarke, J.-P. & Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40, 15–28.